Extracció de dades de IMDB utilitzant tècniques de Screen-Scraping

Treball Final de Grau

Alumne/a (NIU):
Viñas Templado, Adrián (1341050)

Grau d'Estadística Aplicada
UAB - Facultat de Ciències

4 de septiembre de 2017

Agraïments

Al meu tutor del treball, el professor Albert Ruiz Cirera per la seva gran ajuda i colaboració en cada moment de consulta i soport en aquest treball de fi de grau.

Als meus companys de classe per la disposició a ajudar-me en tot moment en el transcurs del treball.

A la meva família per tot el suport rebut durant el grau.

Resum

El creixement i expansió de la coneguda com *red de redes* ens ha portat a l'anomenada era de la informació.
Web scraping és una tècnica utilitzada mitjançant programes de software per extreure informació de llocs web. Usualment, aquests programes simulen la navegació d'un humà en la World Wide Web ja sigui utilitzant el protocol HTTP manualment, o inserint un navegador en una aplicació.

Es presenta a continuació una investigació sobre el desenvolupament de funcions relacionades amb l'*Screen scraping* utilitzant com a software R-project.
Una més relacionada amb la tècnica de screen-scraping purament i l'altra a l'extracció i analisi posterior de les dades.
La mostra ha estat agafada aleatòriament dins del marc mostral d'actors i actrius de la pàgina IMDB per veure si hi han diferències significatives en el nombre de papers protagonitzats en pel·lícules respecte el sexe.

El resultat ha estat que sí hi han diferències significatives entre el nombre de papers que fan els actors de promig respecte les actrius.

S'han aconseguit desenvolupar dues funcions capaces d'extreure i analitzar determinada informació de diverses pàgines web i deixar part del treball enllestit per a que puguin ser modificades o ampliades de codi i extreure més o diferent informació.

Paraules clau
Web scraping, Screen scraping, HTTP, actor, actriu, Rastreator, Inferència, IMDB

Key words
Web scraping, Screen scraping, HTTP,actor, actress, Rastreator, Inference, IMDB

Resumen

El crecimiento y expansión de la conocida como *red de redes* nos ha llevado a la llamada era de la información.
Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando protocolo HTTP manualmente, o insertanto un navegador en una aplicación.

Se presenta a continuacón una investigación sobre el desarrollo de funciones relacionadas con el *Screen-scraping* utilizando como software R-project.
Una más relacionada con la técnica de screen-scraping puramente y la otra con la extracción i análisis posterior de los datos.
La muestra ha sido cogida aleatóriamente dentro del marco mostral de actores i actrices de la página IMDB para ver si hay diferencias significativas en el número de papeles protagonizados en películas respecto el sexo.

El resultado ha sido que sí hay diferencias significativas entre el número de papeles que hacen los actores en promedio respecto a las actrices.

Se ha conseguido desarrollar dos funciones capaces d'extraer i analizar determinada informanción de diversas páginas web y dejar parte del trabajo realizado para que las funciones puedan ser modificadas o ampliadas de codigo y extraer más o diferente información.

Palabras clave
Web scraping, Screen scraping, HTTP, actor, actriu, Rastreator, Inferència, IMDB

Key words
Web scraping, Screen scraping, HTTP,actor, actress, Rastreator, Inference, IMDB

Abstract

The growth and expansion of the known as *network of networks* has led us to the so-called information age.
Web scraping is a technique used to extract information from web sites by software programs. Usually, these programs simulate the navigation of a human on the World Wide Web either using HTTP protocol manually, or inserting a browser into an application.

An investigation is presented about the development of functions related with the *Screen-scraping* using R-project as software.
The first function related to the technique of screen-scraping purely and the second with the extraction and analysis of the data.
The sample has been taken randomly within the sample frame of actors and actresses of the page IMDB to see if there are significant differences in the number of feature films starred acording to the sex.

The result has been that there are significant differences between the number of feature films starred by actors and actresses on average respectively.

It has been possible to develop two functions capable of extracting and analyzing certain information of various web pages and leave some of the work done for the functions can be modified or extended code and extract more or different information.

Key words
Web scraping, Screen scraping, HTTP,actor, actress, Rastreator, Inference, IMDB

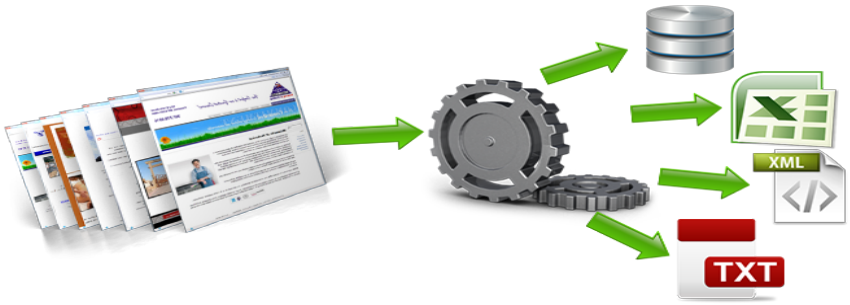
Paraules clau
Web scraping, Screen scraping, HTTP, actor, actriu, Rastreator, Inferència, IMDB

Índice

1. Introducció

A mesura que passen els dies la necessitat de l'accés a la informació augmenta exponencialment. El creixement i expansió de la coneguda com *red de redes* ens ha portat a l'anomenada era de la informació. Tot el que necessitem saber o conèixer el trobem a Internet. Últimes notícies, canvis en el mercat borsari, informació meteorològica, etc. I si es tracta d'una manera senzilla, còmoda i ràpida molt millor.

Pot ocórrer que en determinades ocasions, tan sols vulguem una part del contingut d'una pàgina web, desitjant ignorar la resta de la informació. Per exemple, en una pàgina de meteorològia, únicament volem conèixer el temps actual de la ciutat on ens trobem. I a l'hora, treure algun benefici de les dades que recuperem, és a dir, podriem utilitzar les dades de la situació meteorològica actual per la gestió automàtica de les persianes d'una vivenda intel·ligent. O bé, mostrar la informació al nostre mòbil i, així, conèixer la previsió del temps d'una manera servible.



Aquest procés d'extracció d'una porció d'una pàgina web es conegut com *Screen-scraping*. Segons la Viquipèdia, Screen-scraping es el nom en anglès d'una tècnica de programació que consisteix en agafar una presentació d'una informació per, mitjançant enginyeria inversa, extreure les dades que van donar lloc a la presentació.

Web-scraping és una tècnica de programació per la extracció d'informació de llocs web. Aquests tipus de programes simulen l'exploració humana de la web, ja sigui implementant a baix nivell (HTTP), o inclòs en certs navegadors web, com Internet Explorer o el navegador web Mozilla. Web-scraping està estretament relacionat amb la indexació web: indexa tot el contingut de la Web mitjançant un robot i és una tècnica universal adoptada per la gran majoria dels motors de recerca. Web-scraping es centra en la transformació del contingut web no estructurat, en general en format HTML, en dades estructurades que poden ser emmagatzemades i analitzades en una base de dades local o de full de càlcul. Web-scraping també està relacionat amb l'automatització web, que simula la navegació humana web utilitzant el software d'ordinador. Exemples d'usos de Web-scraping són: comparació de preus en línia, seguiment del temps de dades, detecció de canvis llocs web, la recerca a Internet, la integració de dades Web, etc.

Es tracta d'un procés de recollida automàtica d'informació de la Web, a priori pensada per ser mostrada i no utilitzada per altres sistemes. Web-scraping és un camp amb una evolució àctiva que comparteix un objectiu comú amb la visió de web semàntica, una iniciativa que encara requereix d'avenços en el processament de text, la comprensió semàntica, intel·ligència artificial i les interaccions humà-ordinador. Web-scraping, en canvi, busca solucions pràctiques basades en les tecnologies existents.

2. Context tecnològic

En l'apartat anterior, s'ha esmentat que el camp de Web-scraping està en evolució, existeixen nombroses tecnologies que es podrien haver utilitzat per l'extracció de dades web. No obstant, per decisió pròpia, disseny i implementació, s'ha optat per la que es va a descriure a continuació.

2.1. DOM Parsing

- Què signifiquen les sigles DOM?

Quan un navegador descarrega una pàgina web, crea un model per al document (document object model o DOM en anglès) amb l'estructura de la informació que ha de mostrar.

Anàlisi de DOM al utilitzar els navegadors web, els programes poden recuperar el contingut dinàmic generat per les seqüencies de comandes del client.

Utilitzarem l'enfocament d'anàlisi de DOM durant el desenvolupament d'aquest treball i confiarem en els selectors CSS de la pàgina web per trobar els camps pertinents que contenen la informació desitjada. Però abans de començar, hi ha alguns requisits previs que un necessita amb la finalitat de extreure les dades de forma eficient de qualsevol lloc web.

2.2. Requisits previs

Els requisits previs per a la realització de web scraping en R es divideixen en dos blocs:

· Durant el transcurs d'aquest treball, utilitzarem el paquest *rvest* en R escrit per Hadley Wickham.

Rvest és un paquet nou que facilita la fragmentació de les pàgines web HTML, inspirades en la biblioteca "beautifulsoup". Està dissenyat per analitzar els resultats amb les funcions del paquet *magrittr*, un paquet que implementa la funció % > %, que permet treballar amb redireccionaments (*pipe* en anglès) per a aplicar diferents processos a una mateixa cadena.
Per veure Rvest en acció, imaginem que voldriem extreure informació sobre *TheLego Movie* de IMDB. Comencem per descarregar i analitzar el fitxer amb html():

```
library(rvest)
lego_movie <- html("http://www.imdb.com/title/tt1490017/")
```

Per extreure la classificació, comencem amb *selectorgadget* per esbrinar quin selector de css coincideix amb les dades que desitgem. Usem html_node() per trobar el primer node que coincideix amb aquest selector, extreu els seus continguts amb html_text() i el convertim en numèric amb *as.numeric()*:

```
lego_movie %>%
 html_node("strong span") %>%
 html_text() %>%
 as.numeric()

#> [1] 7.9
```

· A més a més, el coneixement de HTML i CSS serà un avantatge afegit. He observat que la majoria de dades no són gaire sòlides amb els coneixements tècnics d'HTML i CSS. Per tant, utilitzarem un programari de codi obert anomenat Selector Gadget per tal de realitzar scrapping Web. Es una extensió que la proporciona Google Chrome.

Amb això podrem seleccionar les parts de qualsevol lloc web i obtenir les etiquetes rellevants per accedir a aquestes parts simplement fent clic a la part del lloc web.

3. Anàlisi del sistema

Com comentàvem en la introducció d'aquest document, el treball final de grau "*Extracció de dades de IMDB utilitzant tècniques de Screen-Scraping*", pretén aconseguir que la recopilació i presentació de dades concretes, extretes de la pàgina IMDB, es realitzi de forma automàtica i neta per a l'usuari, així com assegurar que la informació està totalment actualitzada.

Ja que el propòsit d'aquest treball és l'estudi de la extracció de dades mitjançant Screen-Scraping s'ha escollit els valors que descriurem a continuació com exemple per demostrar que aquestes tècniques poden aplicar-se a qualsevol camp on hi hagi informació disponible a la xarxa. De la mateixa manera que s'han extret aquests valors, podriem haver escollit diferents camps o valors d'altres pàgines web.

3.1. Objectius i requisits de les funcions

3.1.1. Objectius bàsics de les funcions

En aquest treball hem optat per crear dos tipus de funció amb diferents finalitats, peró alhora, amb la mateixa base; que es basa en el scraping de dades.

La finalitat de la primera funció, anomenada **Rastreator**, serà crear un document .txt donant informació (en aquest cas hem considerat una mini biografia) de l'Actor o Actriu que nosaltres desitgem i que es trobi dins de la base de dades creada a partir de la pàgina de IMDB.

En la nostra segona funció, ens hem centrat una mica més en la part estadística i fa una inferència sobre la mitjana d'actuacions que han tingut els actors i actrius en pel·lícules i analitza si hi han diferències significatives entre els dos sexes respecte al nombre de pel·lícules en mitjana que han participat respectivament.

Posteriorment s'explicarà com aprofitar les funcions desenvolupades per ampliar la informació que ens pot retornar aquestes funcions.

3.1.2. Especificacions i funcionalitats

- La funció Rastreator haurà de crear un .txt amb el nom del actor/actriu escollit i la seva mini biografia corresponent.
- La segona funció ens haurà de determinar si hi han diferències significatives en el nombre d'actuacions de pel·lícules en mitjana respecte al sexe.
- La estructura de les funcions hauran de ser estables i garantir la reusabilitat del codi per poder ser ampliat.

4. Disseny i Implementació

Un cop ha quedat clar quin és l'objectiu de les dues funcions, veurem la seva implementació. Es realitzarà una explicació detallada sobre cada linea de la funció i la seva funció dins d'ella per a que quedi clar en cas de voler ampliar codi dins d'aquesta.

4.1. Funció Rastreator

Primer de tot, abans d'explicar com funciona la funció *Rastreator*, hem creat una funció prèvia anomenada *nameslist* que explicarem a continuació el seu funcionament:

```
nameslist<-function(n){

  star=""
  for (num in seq(1,n,by=1)){


    url=paste("http://www.imdb.com/name/nm",num,sep="")
    pag<-read_html(url)

    actor = pag %%
      html_node(".header") %%
      html_text %%
      strsplit(split = "\n") %%
      unlist() %%
      .[. != ""]

    actor<- trimws(actor)

    actor <- actor[which(actor!="")]
    star <- rbind(star,actor)}
  star=star[-1,]
  star=star[,-2]


  #Para exportarlo a un archivo TXT
  write.table(star, file="Stars.txt")
}
```

Aquesta funció únicament necessita una variable com a entrada, en aquest cas, l'hem anomenada "n" i que voldrà dir el nombre de noms d'actors/actrius que voldrem exportar al nostre .txt.

Llavors, hem creat una variable que es diu "*star*" dins de la funció responsable de guardar els noms exportats de la pàgina IMDB i que tindrà com a longitud (length) **n**. A continuació hem creat un bucle on farà redireccionaments a la pàgina web que li hem escrit, canviant el nombre de la pàgina, es a dir, anirà passant per la pàgina de cada actor/actriu; llegirà el seu nom i després de fer una neteja de text i eliminant línies sense text (implementat a les línies de la 8 a la 12 dins de la funció), exportarà i guardarà el nom del actor/actriu en la variable "*star*" creada al començament de la funció. Un cop ja tenim aquest vector amb tots els noms, la funció crea un .txt on queda registrat tots els noms d'actors/actrius que li podrem demanar a la funció *rastreator* que ens exporti la biografia de qualsevol d'ells.

Un cop tenim aquest .txt anomenat "*Stars*" hem creat una altre funció que la utilizarem dins de la funció principal i que a continuació explicarem quin paper jugarà dins la funció *Rastreator*:

```
position=function(name){

  datos<-read.table("Stars.txt")
  lista=NULL
  for(i in 1:length(datos$x)){
    if(name==datos$x[i]){lista[i]=1}else{lista[i]=0}
  }
  which.max(lista)
}
```

Com el seu nom indica, aquesta funció ens retornarà en quina posició es troba el nom de l'actor o actriu escollit dins del fitxer .txt que hem creat anteoriorment. Únicament li haurem d'entrar a la funció el nom del actor o actriu desitjat i entre cometes.

Aquesta funció obrirà el fitxer i comparará tots els noms que té registrats amb el nom que l'hem introduit a la funció i ens dirà en quina posició es troba.

Un cop explicades les dues funcions anteriors, ara sí procedirem a explicar una de les nostres funcions principals, anomenada ***Rastreator***.

```
Rastreator=function(name){

  pos<-position(name)

  url1=paste("http://www.imdb.com/name/nm",pos,sep="")
  url2=paste("http://www.imdb.com/name/nm",pos,"/bio",sep="")

  pag1<-read_html(url1)

  actor= pag1 %%
    html_nodes(".header") %%
    html_text %%
    strsplit(split = "\n") %%
    unlist() %%
    .[. != ""]

  actor <- trimws(actor)

  actor <- actor[which(actor!="")]


  pag2<-read_html(url2)

  bio<-pag2 %%
    html_node("#bio_content p") %%
    html_text() %%
    strsplit(split = "\n") %%
    unlist() %%
    .[. != ""]

  bio <- trimws(bio)

  bio<- bio[which(bio!="")]

  write.table(t(data.frame(actor,bio)), file="Biography.txt")

}
```

A la funció li podem introduir el nom que volguem del arxiu *Stars.txt*, llavors, la funció Rastreator utilitza la funció previament explicada position on guardará la posició del nom que hem seleccionat.

El següent pas que fa la funció és crear dos redireccionaments a les pàgines web on podrem trobar el nom i la mini biografia de l'actor/actriu desitjat on es guardarà com a variables *url1* i *url2* respectivament. Llavors, aquesta funció llegirà i extraurà el nom i la seva minibiografia.
Aquests dos valors després els ajunta amb un data.frame i l'exporta en format .txt anomenat ***Biography.txt*** ja net i polit.

4.2. Funció d'inferència

Aquesta segona funció està més relacionada amb la part d'estadística del treball, però sense oblidar que el cos del treball és la programació.

Com abans hem fet, explicarem una funció previa que hem creat anomenada *n.total* que ens ajudará a intentar agafar una mosrta de tot el marc mostral que tenim de la pàgina IMDB.
Aquesta funció s'ha creat ja que no sabiem quin era el nombre total d'actors i actrius que tenia la base de dades de IMDB registrats, per tant, el que intenta és esbrinar-ho. A continuació explicarem com ho fa:

```
n.total<-function(link,tol){


  num <- 1

  salt<-2**15

  while(salt > 1){

    passa=0;
    for (i in 1:tol) {
      url=paste(link,num+i-1,sep=""); #print(url);
      if (http_status(GET(url))$reason=="OK"){ passa=1; break;}
    }
    if (passa==1){num=num+salt;
    } else { salt=salt/2; num=num-salt;
    }
  }
  num

}
```

Aquesta funció té com entrada dues variables, una serà el link de la pàgina que volem saber el nombre total de registres, i l'altre la tolerància que estem disposat a assumir. Entenem per tolerància com el nombre màxim de registres buits seguits que estem disposats a deixar passar i continuar amb el recompte, un cop s'excedeixi aquesta tolerància, el programa acabarà i donarà el nombre total de registres.

Com sabem que la pàgina IMDB té molts registres li hem dit al programa que el primer pas, vagi donant salts de mida 2^{15} (sempre es pot modificar). Llavors, fiquem un *while* amb la condició que mentre la variable *salt* sigui més gran a 1, anirà llegint els registres.
La comanda *http_status(GET(url))$reason* ens dirà si la pàgina existeix o per altra banda és buida (no existeix). En cas que existeixi, la variable *passa* augmentarà 1 unitat i la funció passarà a la següent pàgina fent el salt pertinent. En cas que no existeixi la pàgina, divideix la variable *salt* entre dos i aquest serà el valor que utilitzarà per fer el salt. Així correlativament fins que *salt* sigui més petit que 1 com hem dit abans.

Ara si procedirem a explicar la segona funció principal del treball anomenada *final.func.inference*:

```
final.func.inference<-function(n){

  maxim<-n.total("http://www.imdb.com/name/nm",3)

  numb<-sample(maxim,n,replace = FALSE)
  performancemale=NULL
  performancefemale=NULL

  for(i in 1:n){

    url=paste("http://www.imdb.com/name/nm",numb[i],sep="")
    pag<-tryCatch(read_html(url),error = function(e){'Error'})

    if (pag!='Error'){

      creditsmale=pag%%
        html_node("#filmo-head-actor") %%
        html_text() %%
        strsplit(split = "\n") %%
        unlist() %%
        .[. != ""]

      creditsmale <- trimws(creditsmale)

      creditsmale <- creditsmale[which(creditsmale!="")]
      creditsmale<-creditsmale[3]
      if(is.na(creditsmale)=="TRUE"){performancemale[i]<-NA}
      else{
        performancemale[i]<-as.numeric(strsplit(creditsmale,"[^0-9]+")[[1]][[2]])}

      creditsfemale=pag%%
        html_node("#filmo-head-actress") %%
        html_text() %%
        strsplit(split = "\n") %%
        unlist() %%
        .[. != ""]

      creditsfemale <- trimws(creditsfemale)

      creditsfemale <- creditsfemale[which(creditsfemale!="")]
      creditsfemale<-creditsfemale[3]
      if(is.na(creditsfemale)=="TRUE"){performancefemale[i]<-NA}
      else{
        performancefemale[i]<-as.numeric(strsplit(creditsfemale,"[^0-9]+")[[1]][[2]])}
    }


    }
  performancemale<-na.omit(performancemale)
  performancefemale<-na.omit(performancefemale)
  performance<-c(performancemale,performancefemale)

  mmale<-mean(performancemale)
  mfemale<-mean(performancefemale)

  mmale-mfemale
  #Condicions per aplicar el t.test

  #Normalitat
  par(mfrow=c(2,2))
  hist(performancemale,breaks = "Sturges",main="Actor")
  hist(performancefemale,breaks = "Sturges",main="Actress")

  qqnorm(performancemale,main="Actor Q-Q Plot")
  qqline(performancemale)

  qqnorm(performancefemale,main="Actress Q-Q Plot")
  qqline(performancefemale)
```

```
  t.test(x = performancemale, y = performancefemale,
         alternative = "two.sided", conf.level = 0.95)
}
```

Aquesta funció demana una única variable com a entrada anomenada n que serà el nombre de registres que agafará del marc mostral, es a dir, la mida de la mostra la qual farà després la inferència.
A continuació, aplica la funció *n.total* ja explicada anteriorment i guarda en la variable *maxim* el nombre màxim de registres que té la pàgina, en aquest cas, el nombre màxim d'actors i actrius registrats a la pàgina IMDB. Acte i seguit crea un vector de llargada n amb nombres escollits aleatòriament del 1 al nombre màxim sense repetició i el guarda a la variable *numb*; aquesta variable será la mostra d'actors i actrius que utilitzarà per fer la inferència.

Crea dues variables que seran el nombre de papers que han tingut cada actor i actriu de la mostra, però que primerament es buit, i poc a poc s'anirà omplint de dades.

Utilitza el link al qual vol redireccionar-se i aplica la comanda *tryCatch* on la seva funció serà evaluar el codi i assignar controladors d'excepcions quan el codi tingui problemes de connexió amb la pàgina web. A això li afegim un la condició de que si pag és diferent al missatge Error continuarà compilant la funció.

Un cop solucionats els problemes de connexió, la funció aplicarà els selectors CSS corresponents per extreure el nombre de papers que ha tingut l'actor o actriu, i ho guardarà en les variables *performancemale* i *performancefemale* per l'actor o actriu respectivament.

A tot aquest codi, li aplica un bucle de n repeticions (recordem que n ha estat la variable que hem escollit com a entrada de la funció i que representa la mida de la mostra) que pasarà per tots els actors o actrius de la mostra.

Un cop ja tenim els dos vectors que representen els papers dels actors i actrius, li aplica un parell de histogrames i QQ-plots per estudiar la seva normalitat. Per últim, aplica el t.test per a mostres independents per veure si hi han diferències o no en mitjana entre ells i mostra els resultats obtinguts per pantalla.

5. Cas de l'estudi

Un cop ja hem explicat quin és el funcionament del nostre treball, així com s'ha procedit en el seu disseny i implementació, veurem quin es l'aspecte final del mateix, què s'ha desenvolupat i quin és el resultat de la seva execució.

Com hem avançat en els apartats anteriors, la finalitat d'aquest treball és l'estudi de l'extracció de dades web utilitzant tècniques de Screen-scraping.

Com ja sabem, els exemples desenvolupats són:

- Extreure nom i biografia de l'actor o actriu desitjat en format .txt.
- Fer una inferència sobre el nombre d'actuacions d'actors i actrius i veure si hi han diferencies significatives en mitjana.

Els resultats de les funcions dependran del contingut de les pàgines web de les quals desitgem obtenir la informació. Les URL referents als exemples comentats són:

- `http://www.imdb.com`
- `http://www.imdb.com/name/nm0000008/?ref_=nv_sr_1`
- `http://www.imdb.com/name/nm0000008/bio?ref_=nm_ov_bio_sm`

Veurem ara el contingut de les diferents direccions de les quals anem a exportar la informació, i així conèixer quin hauria de ser el resultat esperat de la execució de les funcions.

5.1. Funció Rastreator

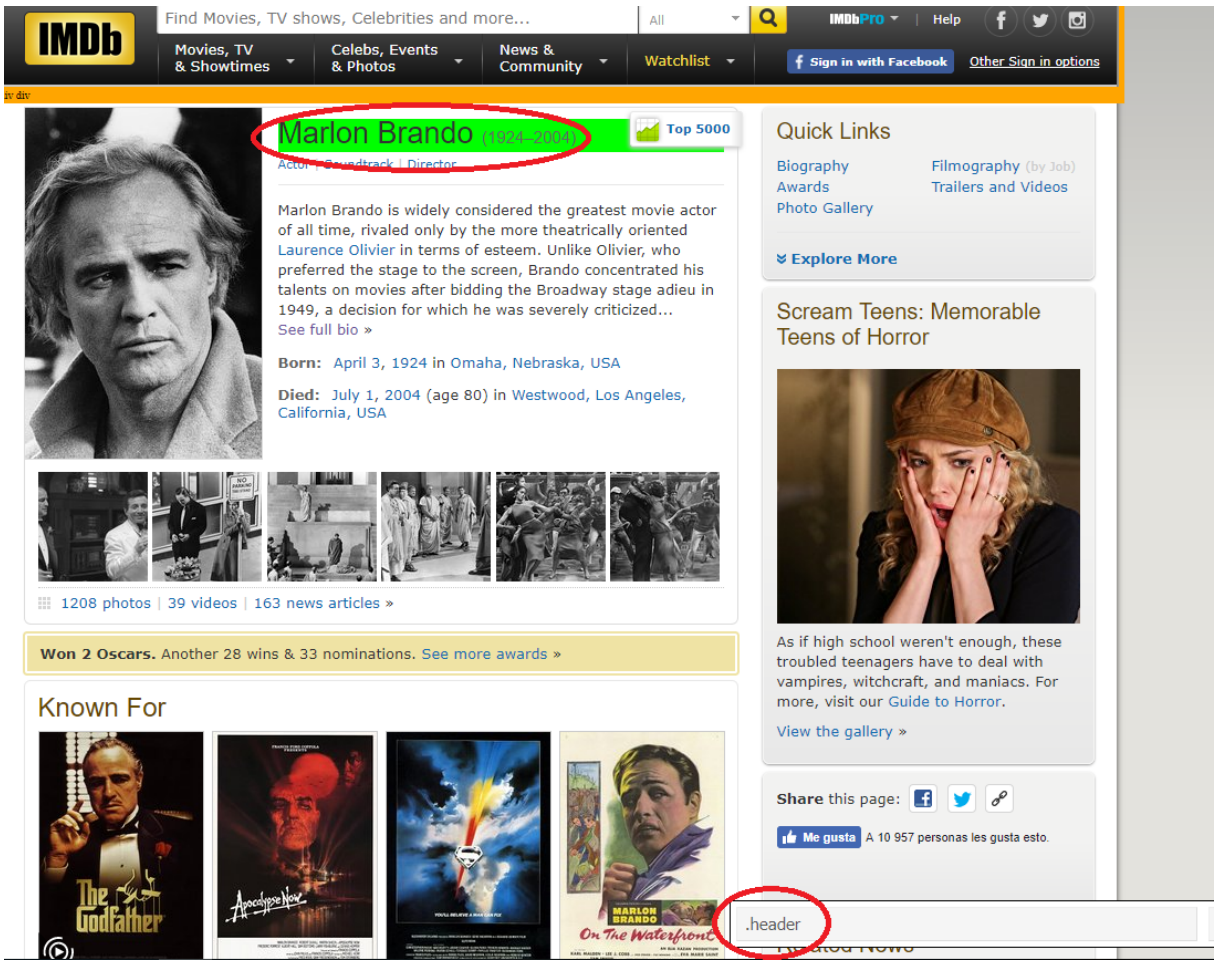
Pel que respecta a la funció de **Rastreator**, volem exportar el nom i la biografia del actor o actriu desitjat. En aquest cas jo he escollit l'actor Marlon Brando que es troba dins del fitxer *Stars.txt* que hem creat anteoriorment.

Stars: Bloc de notas

Archivo Edición Formato Ver Ayuda

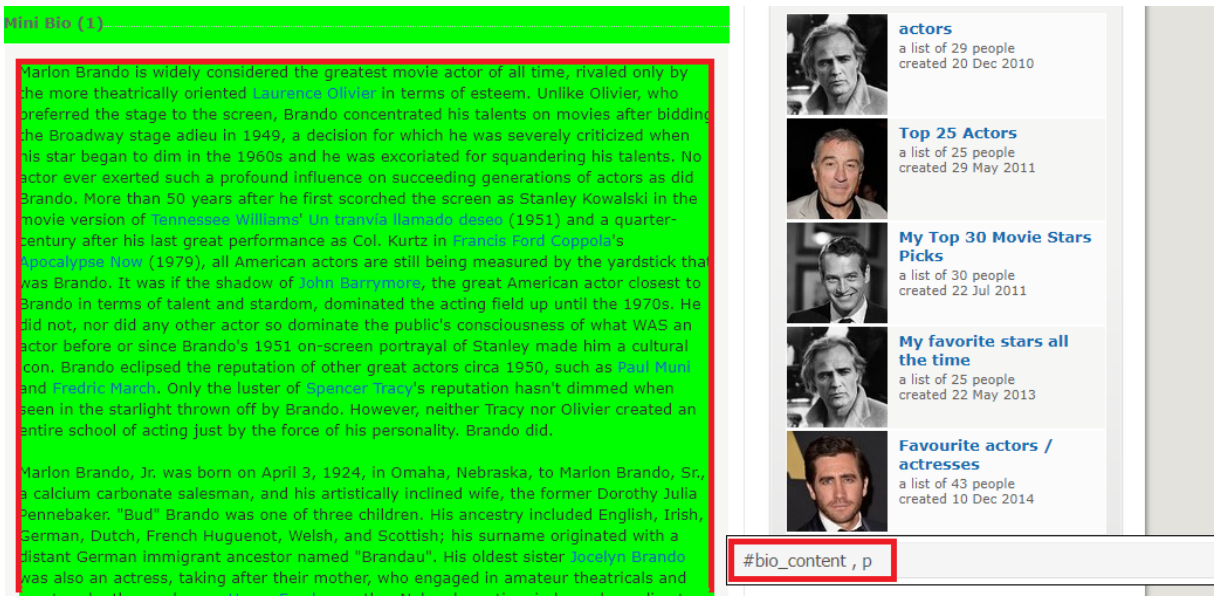
```
"x"
"1" "Fred Astaire"
"2" "Lauren Bacall"
"3" "Brigitte Bardot"
"4" "John Belushi"
"5" "Ingmar Bergman"
"6" "Ingrid Bergman"
"7" "Humphrey Bogart"
"8" "Marlon Brando"
"9" "Richard Burton"
"10" "James Cagney"
"11" "Gary Cooper"
"12" "Bette Davis"
"13" "Doris Day"
"14" "Olivia de Havilland"
"15" "James Dean"
"16" "Georges Delerue"
"17" "Marlene Dietrich"
"18" "Kirk Douglas"
"19" "Federico Fellini"
"20" "Henry Fonda"
"21" "Joan Fontaine"
"22" "Clark Gable"
"23" "Judy Garland"
"24" "John Gielgud"
"25" "Jerry Goldsmith"
"26" "Cary Grant"
"27" "Alec Guinness"
"28" "Rita Hayworth"
"29" "Margaux Hemingway"
"30" "Audrey Hepburn"
```

Per tant, de la pàgina de Marlon Brando (segon Hyperlink), volem exportar el seu nom.



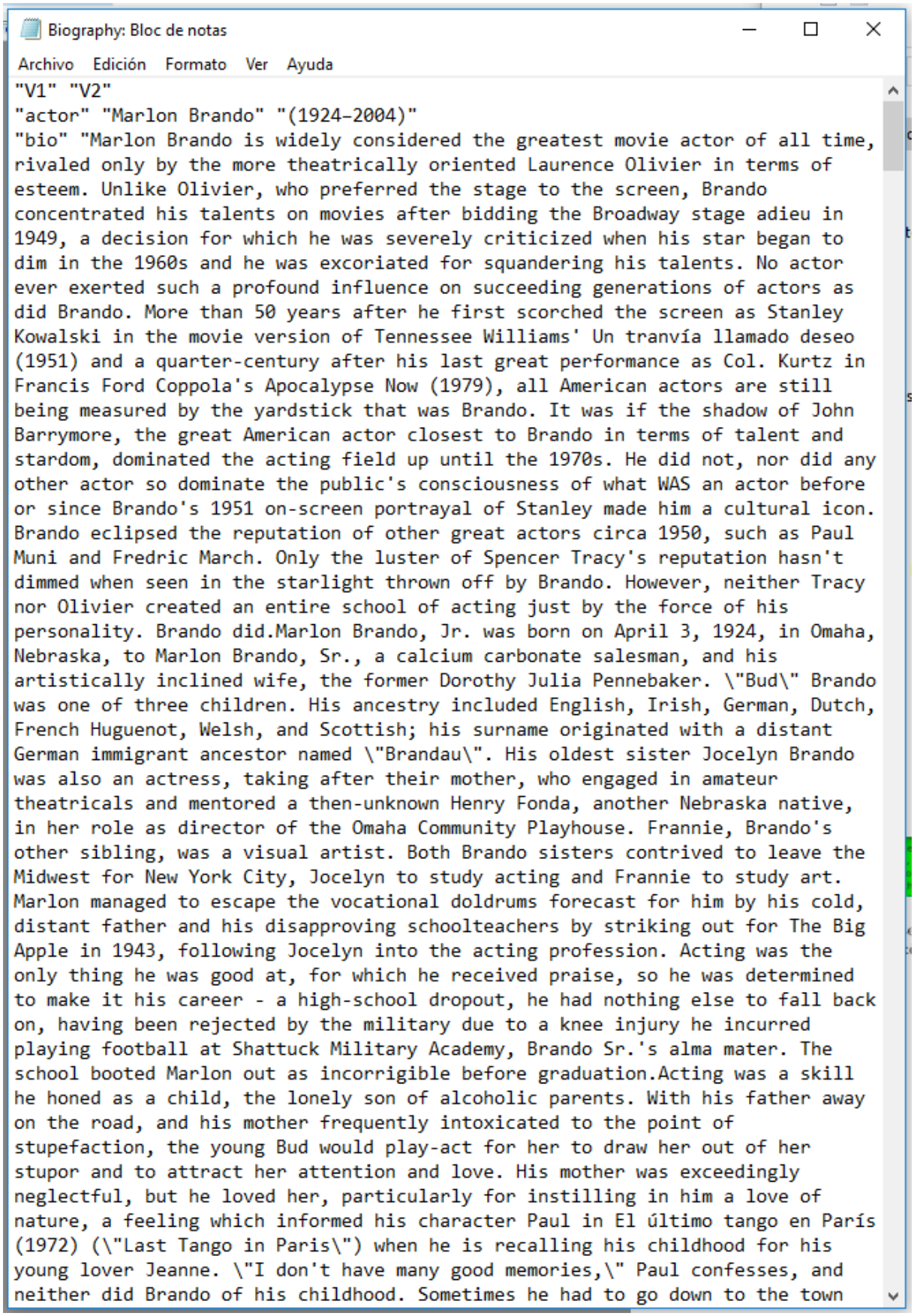
Com veiem, el selector CSS ens diu que el nom de la variable que guarda el missatge nom es diu *.header*, això serà el que haurem de introduir dins de la funció per a que ens llegeixi i exporti el nom.

Seguirem els mateixos pasos per extreure la biografia de l'actor. Veurem com es mostra a la pàgina web:



Com podem veure, el selector CSS en diu que el nom de la variable que guarda aquesta informació és *#bio_ content, p*. Això serà el que introduirem a la funció rastreator.

Per últim, observarem quina és l'exportació final de la nostre funció per veure i comparar-ho amb la pàgina web i així saber si fa el que li estem demanant o no:

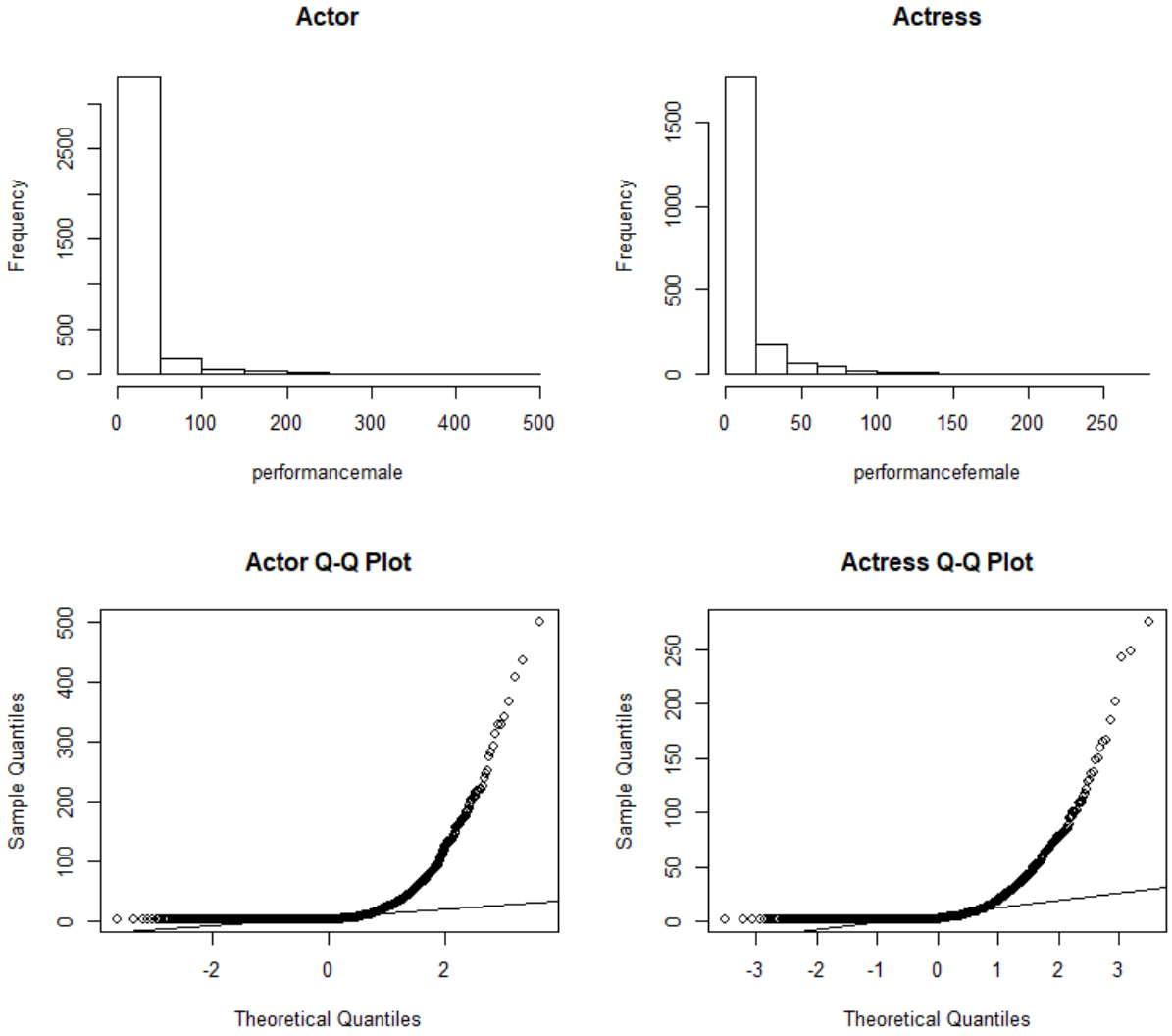

Biography: Bloc de notas
Archivo Edición Formato Ver Ayuda
"V1" "V2"
"actor" "Marlon Brando" "(1924–2004)"
"bio" "Marlon Brando is widely considered the greatest movie actor of all time, rivaled only by the more theatrically oriented Laurence Olivier in terms of esteem. Unlike Olivier, who preferred the stage to the screen, Brando concentrated his talents on movies after bidding the Broadway stage adieu in 1949, a decision for which he was severely criticized when his star began to dim in the 1960s and he was excoriated for squandering his talents. No actor ever exerted such a profound influence on succeeding generations of actors as did Brando. More than 50 years after he first scorched the screen as Stanley Kowalski in the movie version of Tennessee Williams' Un tranvía llamado deseo (1951) and a quarter-century after his last great performance as Col. Kurtz in Francis Ford Coppola's Apocalypse Now (1979), all American actors are still being measured by the yardstick that was Brando. It was if the shadow of John Barrymore, the great American actor closest to Brando in terms of talent and stardom, dominated the acting field up until the 1970s. He did not, nor did any other actor so dominate the public's consciousness of what WAS an actor before or since Brando's 1951 on-screen portrayal of Stanley made him a cultural icon. Brando eclipsed the reputation of other great actors circa 1950, such as Paul Muni and Fredric March. Only the luster of Spencer Tracy's reputation hasn't dimmed when seen in the starlight thrown off by Brando. However, neither Tracy nor Olivier created an entire school of acting just by the force of his personality. Brando did.Marlon Brando, Jr. was born on April 3, 1924, in Omaha, Nebraska, to Marlon Brando, Sr., a calcium carbonate salesman, and his artistically inclined wife, the former Dorothy Julia Pennebaker. \"Bud\" Brando was one of three children. His ancestry included English, Irish, German, Dutch, French Huguenot, Welsh, and Scottish; his surname originated with a distant German immigrant ancestor named \"Brandau\". His oldest sister Jocelyn Brando was also an actress, taking after their mother, who engaged in amateur theatricals and mentored a then-unknown Henry Fonda, another Nebraska native, in her role as director of the Omaha Community Playhouse. Frannie, Brando's other sibling, was a visual artist. Both Brando sisters contrived to leave the Midwest for New York City, Jocelyn to study acting and Frannie to study art. Marlon managed to escape the vocational doldrums forecast for him by his cold, distant father and his disapproving schoolteachers by striking out for The Big Apple in 1943, following Jocelyn into the acting profession. Acting was the only thing he was good at, for which he received praise, so he was determined to make it his career - a high-school dropout, he had nothing else to fall back on, having been rejected by the military due to a knee injury he incurred playing football at Shattuck Military Academy, Brando Sr.'s alma mater. The school booted Marlon out as incorrigible before graduation.Acting was a skill he honed as a child, the lonely son of alcoholic parents. With his father away on the road, and his mother frequently intoxicated to the point of stupefaction, the young Bud would play-act for her to draw her out of her stupor and to attract her attention and love. His mother was exceedingly neglectful, but he loved her, particularly for instilling in him a love of nature, a feeling which informed his character Paul in El último tango en París (1972) (\"Last Tango in Paris\") when he is recalling his childhood for his young lover Jeanne. \"I don't have many good memories,\" Paul confesses, and neither did Brando of his childhood. Sometimes he had to go down to the town

5.2. Funció d'inferència

Primer de tot farem una breu explicació de les condicions que han de cumplir les dades per aplicar un t-test per mostres independents que són les mateixes que pel teorema central del límit:

- Independència: les observacions han de ser independents unes de les altres.
- Normalitat: Les poblacions que es comparen s'han de distribuir de forma normal.
- Igualtat de variància (homocedasticitat): la variància de ambdues poblacions comparades ha de ser la mateixa.

Llavors, la primera condició de independència es compleix ja que les dades han estat agafades de forma totalment aleatòria. També podriem dir que es compleix la tercera condició de igualtat de variàncies.
Per a la segona condició, la funció ens mostra per pantalla l'histograma i el Q-Q Plot de les dues variables:



Els gràfics qqnorm mostren asimetria cap a la dreta i esquerra i els tests realitzats anteriorment troben evidències significatives de que les dades no procedeixen de poblacions amb distribució normal. No obstant, donat que la mida de cada grup es superior a 30, es pot considerar que el t-test segueix sent suficientment robust, encara que es necessari esmentar-ho en les conclusions. En aquest cas, un test no paramètric basat en la mediana (*Mamm-Withney-Wilcoxon test*) o un test de *Bootstraping* serien més adequats. Una altre opció seria estudiar si les dades anòmales son excepcions que es poden excluir del anàlisi.

Un cop es compleixen totes les condicions, podem procedir a fer l'anàlisi:

- **1.Hipòtesi:**

  H_o: no hi han diferències significatives entre les mitjanes del papers protagonitzats entre actors i actrius respectivament.
  H_a: sí hi han diferències significatives.

- **2.Paràmetre estimat (estadístic):**

  Diferència entre les mitjanes mostrals $(X_h - X_d)$.

- **3.Determinació del tipus de test:**

  En el nostre cas, el test d'hipòtesi serà de dues cues.

- **4.Nivell de significació:**

  $\alpha = 0{,}05$

- **5.Càlcul de p-value i comparació amb el nivell de significància:**

  Com les condicions mencionades previament es compleixen, es pot considerar que:

$$T = \frac{(\overline{X_1} - \overline{X_2}) - (\mu_1 - \mu_2)}{SE} \sim t_{(df)}$$

  on

$$SE = \sqrt{\frac{\hat{S_1^2}}{n_1} + \frac{\hat{S_2^2}}{n_2}}$$

  $\hat{S}$ és la quasidesviació típica mostral o desviació típica mostral corregida.

$$pvalue = P(\mid T_{estimada} \mid >= t_{df,1-\alpha/2})$$

```
> final.func.inference(10000)

        Welch Two Sample t-test

data:  performancemale and performancefemale
t = 4.2344, df = 5624.9, p-value = 2.328e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.716894 4.677127
sample estimates:
mean of x mean of y
 14.22793  11.03092
```

- **6.Interpretació:**

  Donat que el p.value és $< 0{,}0001$, disposem de evidències significatives per considerar que existeix una diferència entre el nombre de papers realitzats pels actors i papers realitzats per actrius en mitjana, tot i que cal mencionar que les dades no tenen una distribució normal, sino que s'ha fet una aproximació a la normal.

6. Conclusions

Un cop finalitzat tot el projecte d'aquest treball de fi de grau, ens parem a pensar que conclusions podem extreure de tot el procés que s'ha seguit, des del plantejament inicial fins el final del desenvolupament, és evident que, a diferència d'altres treballs, el principal objectiu d'aquest no ha estat la implementació d'un parell de funcions amb una funcionalitat extensa i dirigida a un usuari final que pugui utilitzar les funcions freqüentment, sino que l'objectiu més important és l'aprenentatge de la tècnica de Screen-scraping que s'ha anant explicant al llarg del document.

Com ja s'ha explicat, el procés d'aprenentatge ha condicionat d'una manera molt forta el desenvolupament del treball, és a dir, s'ha anant implementant codi a mesura que s'anava aprenent sobre el funcionament de la tècnica Sreen-scraping. Primer de tot es va tractar amb exemples senzills que servien d'aprenentatge, i un cop s'havia dominat la tècnica, mes o menys, s'aplicaven codis una mica més elaborats i complexes. Un cop vam tenir el disseny clar, es va implementar tota la resta de codi.

S'ha aconseguit desenvolupar dues funcions capaces d'extreure i analitzar determinada informació de diverses pàgines i deixar part del treball enllestit per a que puguin ampliar el codi i extreure informació d'altres pàgines web.

Bàsicament, la meva tasca com futur estadístic, ha estat l'aprenentatge de, per a mi, una nova tècnica d'extració de dades de la Web dins del software que més he utilitzar al grau i nous mètodes de treball (es planteja un problema que l'ha de resoldre un mateix), així com totes les tasques d'anàlisi i disseny, que després de l'aprenentatge, han estat en les que més temps s'ha invertit.

7. Limitacions i prospectiva

L'Screen scraping pot ser difícil si no es tenen les eines adequades. En gran mesura, està completament a mercè del lloc web objectiu i aquest pot canviar en qualsevol moment, sense previ avís. O bé, pot contenir un JavaScript defectuos que provoca que es bloquegi i mostri un comportament extrany. El servidor que allotja el lloc web pot bloquejar-se o sometre's a un manteniment.

Altres dificultats, que poden arribar fins i tot a posar-te limitacions seria extreure dades de llocs web molt complexes, llocs on demanen un registre previ, grans extraccions de dades o extracció de dades que no són HTML.

Un pas futur podria ser la implementció d'un paquet de R-project, en aquest cas, enfocada totalment als usuaris que vulguin utilitzar aquesta tècnica anomenada Screen-scraping i que els hi fos molt més còmode l'extracció de dades únicament entrant l'HTTP de la página o pàgines i el o els CSS que volen extreure.

8. Bibliografia

FUNDACIÓN Wikimedia. Web Scraping. Actualizada: 10 mayo 2017. Disponible en: https://es.wikipedia.org/wiki/Web_scraping

SAURAV Kaushik. Beginner's Guide on Web Scraping in R (using rvest) with hands-on example. Actualitzada: March 21, 2017. Disponible en: https://www.analyticsvidhya.com/blog/2017/03/beginners-guide-on-web-scraping-in-r-using-rvest-with-l

SKIN-Strtrans-Syntoc. Web Scraping. Disponible en: https://contentgrabber.com/Manual/introduction.htm

PÀGINA web IMDB. Disponible en:http://www.imdb.com/

9. Annex

Codi R utilitzat des del començament del treball fins al final:

```
 library(rvest)
library(httr)

 film<-paste("http://www.imdb.com/name/nm1/")


director= film %>%
  html_nodes(".header") %>%
  html_text %>%
  strsplit(split = "\n") %>%
  unlist() %>%
  .[. != ""]

director <- trimws(director)

director <- director[which(director!="")]


pag<-read_html(film)
credits=pag%>%
  html_node("#filmo-head-actor") %>%
  html_text() %>%
  strsplit(split = "\n") %>%
  unlist() %>%
  .[. != ""]

credits <- trimws(credits)

credits <- credits[which(credits!="")]
credits<-credits[3]
credits<-as.numeric(strsplit(credits,"[^0-9]+")[[1]][[2]])
credits


##################Funcib exportaci###################################
nameslist<-function(n){

  star=""
  for (num in seq(1,n,by=1)){


    url=paste("http://www.imdb.com/name/nm",num,sep="")
    pag<-read_html(url)

    actor = pag %>%
      html_node(".header") %>%
      html_text %>%
      strsplit(split = "\n") %>%
      unlist() %>%
      .[. != ""]

    actor<- trimws(actor)
```

```
    actor <- actor[which(actor!="")]
    star <- rbind(star,actor)}
  star=star[-1,]
  star=star[,-2]



  write.table(star, file="Stars.txt")
}



 position=function(name){

  datos<-read.table("Stars.txt")
  lista=NULL
  for(i in 1:length(datos$x)){
    if(name==datos$x[i]){lista[i]=1}else{lista[i]=0}
  }
  which.max(lista)
 }

 Rastreator=function(name){

  pos<-position(name)

  url1=paste("http://www.imdb.com/name/nm",pos,sep="")
  url2=paste("http://www.imdb.com/name/nm",pos,"/bio",sep="")

  pag1<-read_html(url1)

  actor= pag1 %>%
    html_nodes(".header") %>%
    html_text %>%
    strsplit(split = "\n") %>%
    unlist() %>%
    .[. != ""]

  actor <- trimws(actor)

  actor <- actor[which(actor!="")]


  pag2<-read_html(url2)

  bio<-pag2 %>%
    html_node("#bio_content p") %>%
    html_text() %>%
    strsplit(split = "\n") %>%
    unlist() %>%
    .[. != ""]

  bio <- trimws(bio)

  bio<- bio[which(bio!="")]
```

```
  write.table(t(data.frame(actor,bio)), file="Biography.txt")

}

Rastreator("Marlon Brando")

####################################

n.total<-function(link,tol){


  num <- 1

  salt<-2**15


  while(salt > 1){

    passa=0;
    for (i in 1:tol) {
      url=paste(link,num+i-1,sep=""); #print(url);
      if (http_status(GET(url))$reason=="OK"){ passa=1; break;}
    }
    if (passa==1){num=num+salt;
    } else { salt=salt/2; num=num-salt;
    }
  }
  num

}


final.func.inference<-function(n){

  maxim<-n.total("http://www.imdb.com/name/nm",3)

  numb<-sample(maxim,n,replace = FALSE)
  performancemale=NULL
  performancefemale=NULL

  for(i in 1:n){

    url=paste("http://www.imdb.com/name/nm",numb[i],sep="")
    pag<-tryCatch(read_html(url),error = function(e){'Error'})

    if (pag!='Error'){

      creditsmale=pag%>%
        html_node("#filmo-head-actor") %>%
        html_text() %>%
        strsplit(split = "\n") %>%
        unlist() %>%
        .[. != ""]

      creditsmale <- trimws(creditsmale)
```

```
      creditsmale <- creditsmale[which(creditsmale!="")]
      creditsmale<-creditsmale[3]
      if(is.na(creditsmale)=="TRUE"){performancemale[i]<-NA}
      else{
        performancemale[i]<-as.numeric(strsplit(creditsmale,"[^0-9]+")[[1]][[2]])}

      creditsfemale=pag%>%
        html_node("#filmo-head-actress") %>%
        html_text() %>%
        strsplit(split = "\n") %>%
        unlist() %>%
        .[. != ""]

      creditsfemale <- trimws(creditsfemale)

      creditsfemale <- creditsfemale[which(creditsfemale!="")]
      creditsfemale<-creditsfemale[3]
      if(is.na(creditsfemale)=="TRUE"){performancefemale[i]<-NA}
      else{
        performancefemale[i]<-as.numeric(strsplit(creditsfemale,"[^0-9]+")[[1]][[2]])}
    }


    }
  performancemale<-na.omit(performancemale)
  performancefemale<-na.omit(performancefemale)
  performance<-c(performancemale,performancefemale)

  mmale<-mean(performancemale)
  mfemale<-mean(performancefemale)

  mmale-mfemale



  par(mfrow=c(2,2))
  hist(performancemale,breaks = "Sturges",main="Actor")
  hist(performancefemale,breaks = "Sturges",main="Actress")

  qqnorm(performancemale,main="Actor Q-Q Plot")
  qqline(performancemale)

  qqnorm(performancefemale,main="Actress Q-Q Plot")
  qqline(performancefemale)


  t.test(x = performancemale, y = performancefemale,
         alternative = "two.sided", conf.level = 0.95)
}

final.func.inference(10000)
```